UM-400

組み込み用/製品版・超音波モーターコントローラ

ユーザーズ・マニアル

株)ラボラトリ・
イクイップメント・コーポレーション
http://www.labo-eq.co.jp
〒300-0034 茨城県土浦市港町 1-7-3
TEL 029-821-6051
http://www.nabe-e.com Y.Tanabe

2013/01/24

目次

内容

はじめに 4
UM-400M の取扱説明 5
UM-400P の取扱説明 6
リミッタの接続 7
組み込み方法(ハードウェアの接続例) 8
D6060E D6060S ドライバ内部接続図 (ケース製品の内部) 9
製品版とモーターの接続方法 10
USR30 モーターの接続例(製品版ケース間の接続) 11
パソコンの準備 12
試験ソフトウェアを起動しましょう (USB バージョンのみ) 13
画面表示の説明 15
操作時の制限事項 17
本体ボタン操作 17
プログラミング TOOL について 18
UM400DLL.DLL の使い方 19
UM400DLL 解説(1) 専用 DLL ライブラリ命令 20
USB インターフェイスの OPEN 処理(1) パラレル型 USB チップ製品の OPEN 20
USB インターフェイスの OPEN 処理(2) シリアル型 USB チップ製品の OPEN 20
USB インターフェイスを CLOSE する 21
USB 通信の基本命令 WRITE 22
USB 通信の基本命令 READ 22
USB 通信の基本命令 受信バッファデータ数の取得 23
USB 通信の基本命令 送受信バッファのサイズを指定する。 23
UM-400 専用命令 バージョン情報の読み出し 24
UM-400 専用命令 デバッグモードの ON 25
UM-400 専用命令 デバッグモードの OFF 25
UM-400 専用命令 モータの緊急停止 26
UM-400 専用命令 モータ設定情報をハードウェアに記録 27
UM-400 専用命令 データム処理 28
UM-400 専用命令 モーター回転速度の指定 29
UM-400 専用命令 リミットの極性の設定 30
UM-400 専用命令 絶対座標位置へ移動する 31

UM-400 専用命令　　指定した座標分、相対移動する........32
UM-400 専用命令　　座標値を変更します。........33
UM-400 専用命令　　座標値を読み込みます。........34
UM-400 専用命令　　モータの状態、リミッタ状態を読み出します。........35
UM400DLL 解説(2) USB 命令........36
USB 命令の FORMAT........36
USB / RS32C 回線上の命令内容........37
メンテナンス、初期化処理........38
UM-400P ボードを初期化する........39
UMー400リミッタの極性を指定する........40
制御モータ数の変更........41

マニアル変更履歴

03-JUN-2010	初版
02-JUL-2010	納入ソフトの修正とメンテナンス操作を追加
25-JUL-2012	エンコーダ、リミッタスイッチ入力回路解説部分を追加
02-JUL-2012	ケース製品版のモーター接続コネクタ解説を追加
02-AUG-2012	試験ソフトのデータム処理の指定方法を変更した。
21-AUG-2012	RS232C　インターフェイスバージョンのサポート
13-AUG-2012	USB / RS232C 回線データ FORMAT　を追加
24-JAN-2013	D6060S 内部接続図等の追加と変更、

はじめに

UM-400 は、株式会社新生工業から販売されている、超音波モータを位置決めして、USB 接続で制御するコントローラです。

コントロール可能な超音波モータ

USR30 型モータと新生工業ドライバのセット

(写真は、新生工業のホームページから)

USR60 シリーズとドライバのセット
USR30 または USR60 モータのセットは新生工業から購入します。
当社から、モータとドライバを購入して納品することもできます。

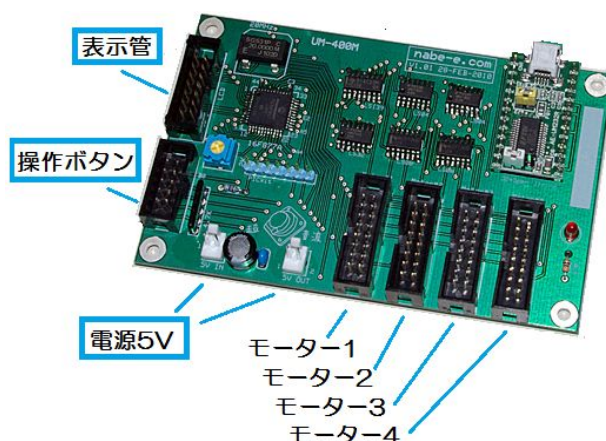
(写真は、新生工業のホームページから)

制御コントローラ基板は二つの基板から構成されています。
UM-400M　４軸モータ制御基板　UM-400P を最大４枚接続できる
UM-400P　モータ制御基板

簡単な取り付け金具付きの組み込みタイプ(表紙)と UM-400 専用ケースに収納されたタイプがあります。
専用ケースの場合は、２軸用と、４軸用のケースとなります。
(専用ケース付きの場合は、別途費用がかかります)

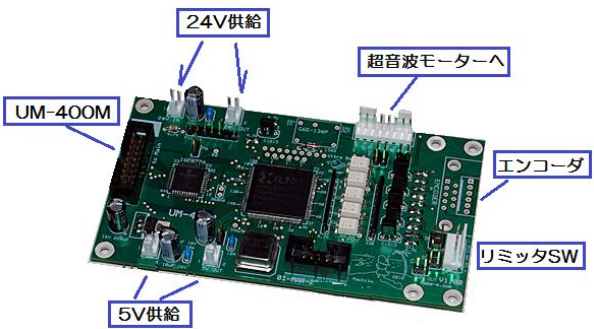
UM-400M の取扱説明

USB インターフェイスを搭載した
4 軸モータコントローラです。
モータ 1 から 4 まで 4 枚の
UM-400P 基板を接続できます。

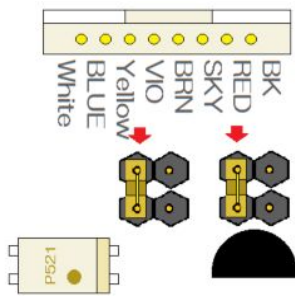
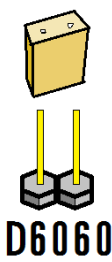
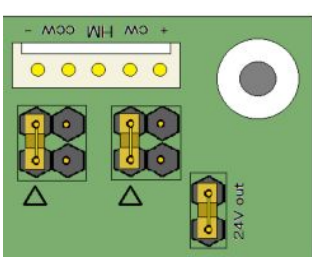
操作ボタンと、表示管は
出荷時の初期設定のために使います。
初期化は、当社出荷時に設定しています。
購入後、モータを追加した場合などに、接続モータ数を変更したり、リミッタスイッチの
電気的な極性を変更するときに、使います。
詳しい使い方は、後述します。

基本的に、UM-400M は USB インターフェイスからのリモートコマンドで動作するるよ
うに設計してあります。
USB インターフェイスには、ユニークな 8 文字のコードが割り振られています。
このコードは、UM-400M の基板に張り付けていります。大文字小文字を区別するコードです。
リモートソフトウェアから、USB インターフェイスでアクセスする場合、このコードを指定してコントロールします。　つまり、このユニークなコードにより、複数の UM-400M をリモートコントロールして、一度に沢山の超音波モータを制御できます。

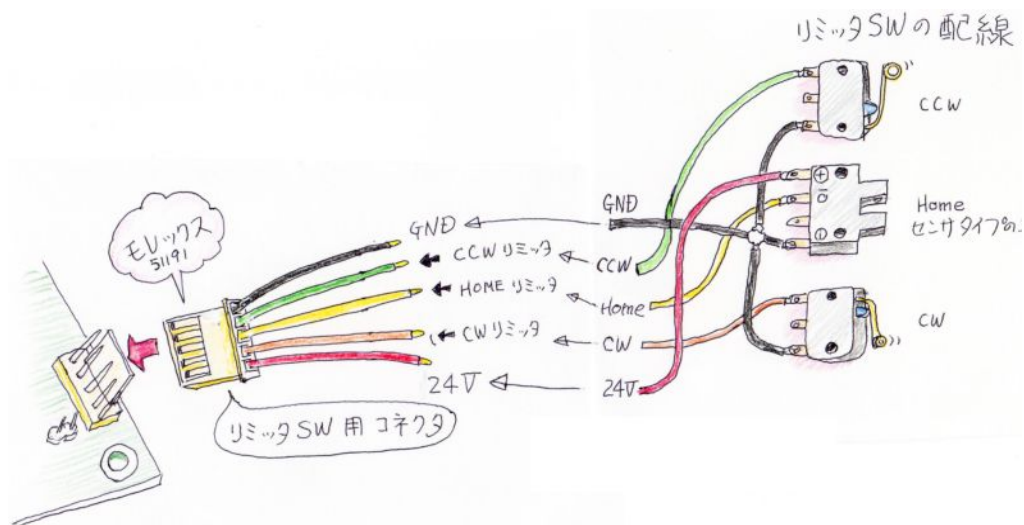
RS232C インターフェイス搭載も可能です。
　RS232C 19200bps パリティ無、Stop=1、8bit で制御可能で接続するバージョンも用意できます。　購入時に指定してください。

UM-400P の取扱説明

UM-400P は、個々のモータのコントロールを行います。
UM-400M へ 16 芯のフラットケーブルで接続します。

接続するコネクタの説明

5V 供給	ロジック用電源を供給します。　どちらかのコネクタから供給
24V 供給	超音波モータの D603 や D6060E などの電源を供給します。
HEAD16	UM-400M へフラットケーブルで接続します。 モータの回転方向をジャンパーピンで反転させることができます。
SIP8 芯	超音波モータドライバとつなぎます(モレックス)
エンコーダ	D6060E の場合は DSUB-9 コネクタ、D6030 の場合は モレックスの６ピンコネクタが付けてありますからエンコーダまたは ドライバ回路と直接配線してください。
リミッタ SW	(+)ccw と(-)cw 方向、Home 位置リミッタを接続します。 リミッタ SW は基板の上で(+)と(-)の反転できます。 機械式リミッタの場合、電源供給(24V)をジャンパピンで制限できます。

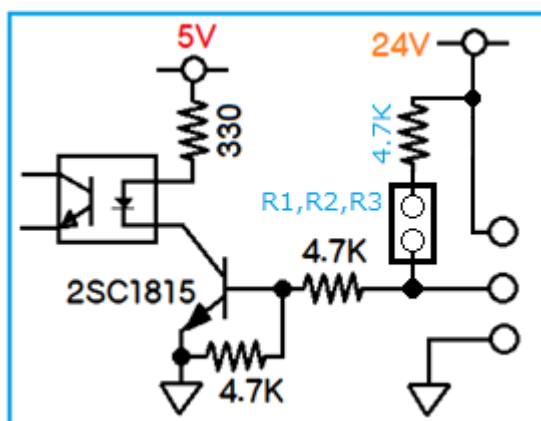
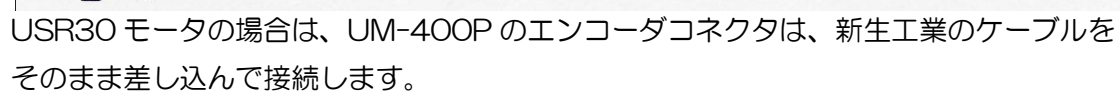
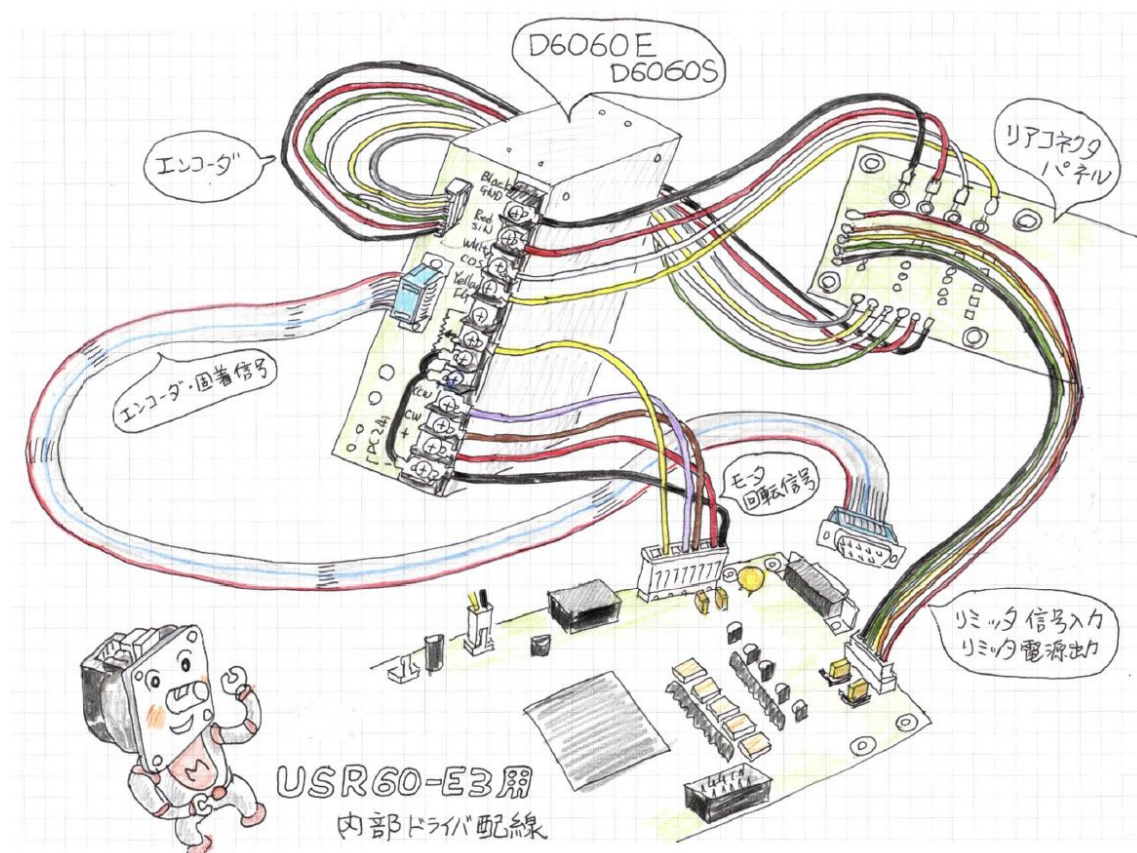
モータ回転方向、納入時設定

リミッタ SW 方向納入時設定

納入時は、モータの機械的な回転方向(正面から見て時計の針の回転方向がプラス方向)に設定してあります。実際のモータの回転方向の正負の方向とリミッタの方向は別々に設定できます。
基板中央左側にはモーター選択ジャンパーがあります。 USR30 モータの場合は解放しナＵＳＲ６０，Ｄ６０６０Ｅドライバの場合はジャンパーピンで短絡します。

リミッタの接続

機械式のマイクロスイッチや、フォトセンサータイプのセンサーを利用できます。
電源が必要なセンサーを使う場合は、電源 24V で動作可能なセンサを利用してください。
コントローラに　24V OUT　のジャンパーがあります。
電源を必要としないセンサを使う場合は、24V 出力のジャンパーを外して使います。
センサーからの情報 CW,HOME,CCW がオープンコレクタの場合は、コントローラー側で 4.7KΩのプルアップをジャンパーR1,R2,R3 で接続できます。
R１は CW 信号、R2 は HOME 信号、 R3 は CCW 信号に対応しています。

電源ショートの場合の簡単な、接点復帰できるヒューズ部品がついていますが
24V の電源を使う場合は、注意してください。
納入時には、リミッタスイッチの代りに、トグルスイッチの付いた、小さな基板が接続されています。
スイッチは上側で短絡している
負論理接続です。

リミッタ入力回路は、右図のようです。
24V の電源､GND と信号入力があります。
24V 以外の電気仕様のセンサの場合は
利用者側で電源を用意してください。
センサー出力は、回路図で対応可能な
電気仕様で接続してください。

組み込み方法(ハードウェアの接続例)

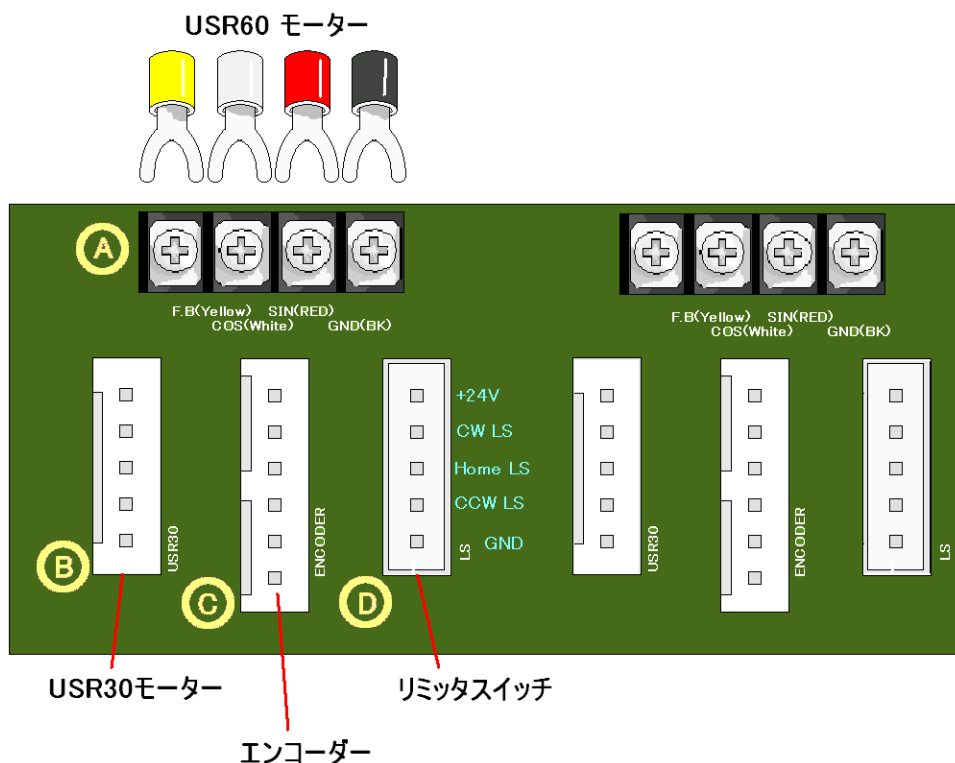
USR30 モータの場合は、UM-400P のエンコーダコネクタは、新生工業のケーブルをそのまま差し込んで接続します。

D6060E D6060S ドライバ内部接続図　(ケース製品の内部)

ドライバの下から(1)-(5)-(6)はグランドですから、図の様に共通につなぎます。

当社からケース製品として納入した場合は、図のようなカラーケーブルで配線してあります。

小型の USR30E モーター(エンコーダ回路付)の場合は、エンコーダ信号は、ドライバではなくコントローラ基板に直接接続します。

製品版とモーターの接続方法

製品版の場合、新生工業のモーターを直接、UM-400 筐体のコネクタへ差し込みます。

筐体のリアパネルのコネクター

ケーブル側のコネクタ仕様

	メーカー	形名	RS 品番
A	ニチフ	GMEV1.25 相当品	
B	モレックス	51191-0500 Pin=50802-8100	480-7625
C	モレックス	51191-0600 Pin=50802-8100	480-7631
D	日本圧着端子製造	XHP-5 Pin=BHF-001T-0.8BS	353-1658

購入時の USR60E のモーターケーブルには、圧着端子で仕上げてあります。
USR60 モータの場合は(A)のコネクタへケーブルの色を合わせてねじ止めします。
USR30 モータも購入時のケーブルをそのまま、接続します。
リミッタスイッチのケーブルは、日圧 XHP-5 ハウジングで作成します。
リミッタ用ケーブルは、モーターケーブルと同じ長さのケーブルを納品しています。
実際のリミッタへの配線を行う場合は、納入したケーブルを利用すると、コネクタの作成を省けます。

USR30 モーターの接続例(製品版ケース間の接続)

USR60 モーターの接続例(製品版ケース間の接続)

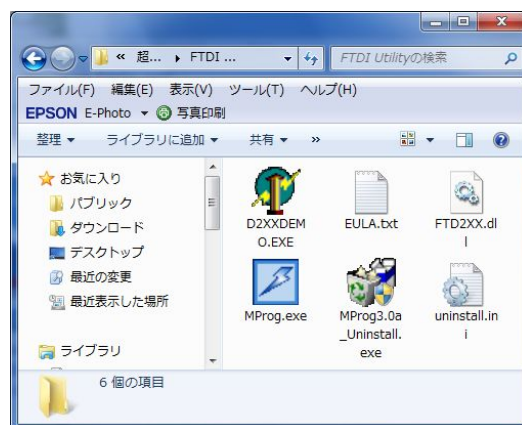
パソコンの準備

納品した CD には、Windows 用のソフトウェアが入っています。
納入した超音波モータボードにモータや電源、ドライバを接続して電源を入れて
問題なく LCD 表示が表示されれたならば、USB ケーブルで、パソコンとコントローラ
の USB コネクタをつないでください。

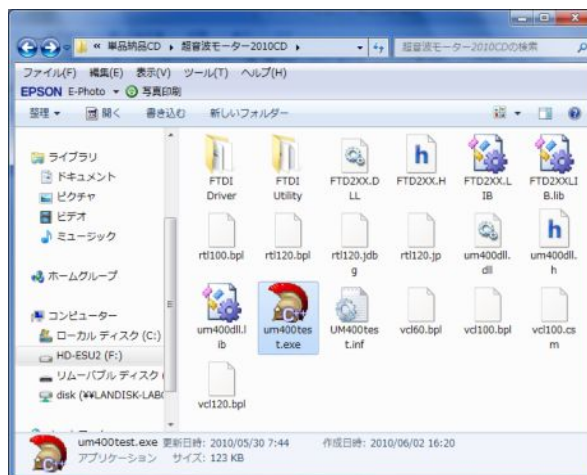
ドライバのインストールが始まります。ドライバファイルの要求が出たら
ドライバの定義ファイルを、CD 内　FTDI Driver フォルダを指定してインストールを
続けてください。

最新のドライバをインストールしたい場合は
http://www.ftdichip.com/ の Drivers に入り、ダウンロードして使ってください。

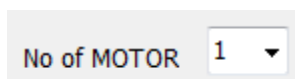
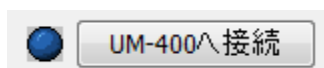
USB が接続できたら
CD 内の　FTDI Utility の中に入り
D2XXDEMO.EXE を実行してみてください。

超音波モータコントローラに取り付けてある FTDI 社のチップのシリアル番号が
表示されます。
当社であらかじめ、シリアル番号を調べて
コントローラ基板、ケース付きで納入した
場合は、ケースの USB コネクタ付近に
シリアル番号のテープが貼ってありますので、同じ番号であるか確認してください。

シリアル番号の意味
同じ超音波モーターコントローラを複数台使って 4 個以上のモータを制御する場合に
このシリアル番号はユニークなので、排他制御ができます。
当社の試験ソフトも、接続する前に、このシリアル番号を指定してから接続するように
作ってあります。

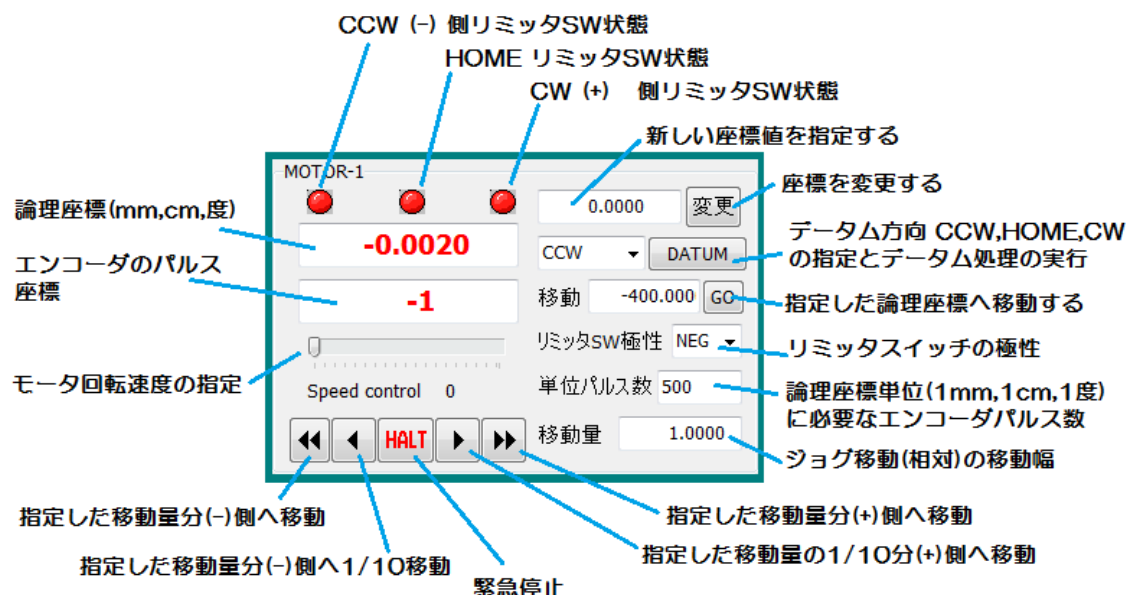
試験ソフトウェアを起動しましょう　(USB バージョンのみ)

CD の中の
UM400test.exe
を起動してください。

最初に起動したときは、UM-400 S.N に　12345678 と表示されます。
ここに、正しいシリアル番号を入力してください(大文字と小文字は区別します)

UM-400 S/N　A9009psS

次に、接続しているモーターの数を指定します。
モータの接続数は、UM-400M の基板へも設定する必要があります。
出荷時にこの数値は、初期設定(後述)で、接続したモーター数を設定しています。
モータを増設した場合は、後述するメンテナンスの説明を読んで、変更します。

モータの数(1,2,3,4)を指定したら、これで超音波モータと USB で接続できます。

上の UM-400 へ接続　のボタンをクリックします。

左下に REV3.08 のようにリビジョン番号が表示されて、オンライン表示になります。

画面表示の説明

パルス座標: 超音波モータに付いているエンコーダのパルスを計数して、座標を表している
　　　　　座標を、ここでは絶対座標と呼んでいます。
論理座標:　実際の長さや角度の単位に換算した座標を、ここでは論理座標と呼んでいます。

エンコーダの精度:
　　　　　取り付けてあるエンコーダは、一回転で 500 または 1000 個のパルスを
　　　　　出力します。
　　　　　現在の超音波モータコントローラは、エンコーダのそのままの計数で座標を決定
　　　　　していますから、一回転で 500 ないし 1000 パルスのパルス座標が変化します。
　　　　　(将来的には、エンコーダ計数方式を変更して、1000、2000 パルス/回転に
　　　　　　機能 UP する計画があります)
リミッタスイッチの極性:
　　　　　通常は NEG で使うのが一般的です。
　　　　　リミッタがついていない場合は POS とすると、リミッタの配線をしなくても
　　　　　モーターは回ります。
　　　　　試験プログラムを最初に起動した場合は、NEG/POS の指定を NEG に再設定
　　　　　してください(NEG でも重ねて NEG を指定してください)
　　　　　出荷時に、コントローラ基板は NEG に設定して出荷しています。

座標範囲:　パルス座標の範囲は、32 ビット整数です。
ただし、エンコーダの内部の計数は 24 ビットです、座標を変更すると、
その位置を中心に±23 ビット幅まで移動できます。
-8388608　から　+8388607 の範囲でモータを回すことができます。
この移動範囲は、座標を変更した位置からですから、838 万を超える位置へ
移動する場合は、一旦座標ほ更新することで、さらに 838 万パルスまで
移動可能です。
モータ速度: スライダーを移動して 0 から 255 までの速度が指定できます。
スライダーは AD 変換回路で 0V から 4V 程度の電圧を出力します。
超音波モーターの速度コントロールは　0.5V 程度から 3.5V の範囲で
変化していますので、スライダーの左右の端付近では、速度の変化は少ないです。
サンプルプログラムでは、モータスピードの設定が起動時には一番遅い速度に
なっています。　スライダーを一度動かして、速度を変更することで
スライダーの指定速度になります。

操作時の制限事項

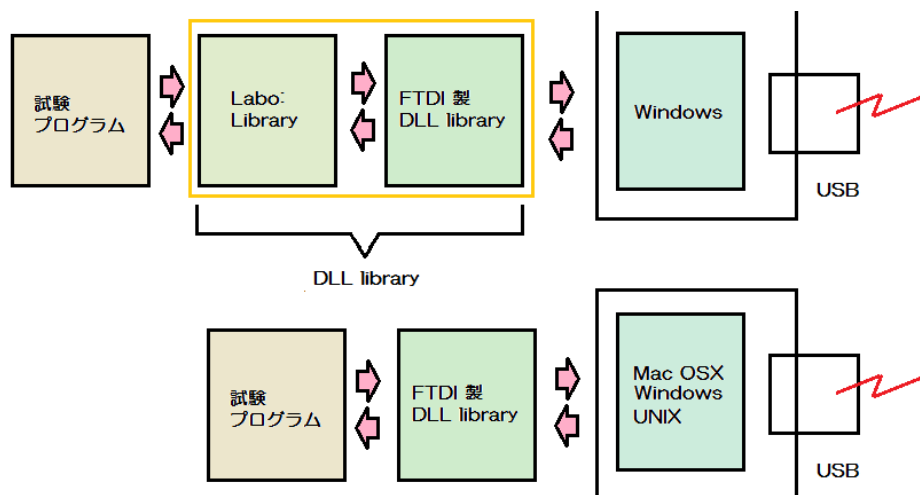
コンピュータ側からリモート制御を開始した場合は、本体のボタン操作はしないようにしてください。
リモート中にボタン操作をすると、通信がとまってしまうことがあります。
将来的には、リモート命令を受けたら、ボタン操作は、電源を再投入するまで、機能しないようにする様に変更する予定です。

本体ボタン操作

本体のボタンは、メンテナンス用です。
新しいモータを追加した時に、そのモータのための初期値(リミッタスイッチ極性など)を設定するためなどに使います。
LCD 表示管は、メンテナンス時に、確認するために使います。

ご自分で、ソフトウェアを開発される場合は、リモート命令に、デバッグモード命令があり、デバッグモード ON にすると、パソコンからの通信データを本体の LCD へ表示させることができます。

プログラミング TOOL について

UM-400 をコントロールするためのソフトウェアツールは以下のような構造になっています。

USB 用 IC は FTDI 社のチップを使っています。
FTDI 社は様々な OS のライブラリを提供しています。当社は Widows 用のライブラリを利用して、超音波コントローラに特化した命令を追加した DLL ライブラリを作成して提供しています。
FTDI 社のライブラリのみを使ってプログラミングすることもできます。
上の図は、
1）FTDI 社のライブラリを含んだ当社の DLL ライブラリ（UM400DLL.DLL）
　　を使ってプログラムを開発する場合。
2）直接プログラムから FTDI 社の DLL ライブラリを直接呼び出す場合。

のイメージ図です。
当社の UM400DLL.DLL を利用する場合は、Windows の開発言語だけで利用できます。

UM400DLL.DLL の使い方

UM400DLL.DLL は UM-400 用に特化した DLL ライブラリです。
Windows で動作するコンパイラ VC++、C++Builder、LabVIEW などの言語で利用できます。
通常、VC++や C++Builder の場合、直接言語から DLL ライブラリを呼び出す方法ではなく、import library -.lib を使って呼び出します。
当社が納品するCDは、C++Builderを利用しているため、C++Builder用のimport library が入っています。
C++Builder 以外の import library を利用する言語を使う場合は、それぞれの言語のユーティリティで import library を生成してください。

FTDI 社の提供する、TOOL のみでソフトウェアを開発する場合は
後述する、UM-400 基本命令仕様　を参考にして、TOOL を作成してください。

FTDI 社の USB 用 IC には二つの種類があります。
1）シリアル通信チップ
3）パラレル通信チップ

UM-400 装置は、シリアル通信チップを使っています。
UM400DLL ライブラリはこのチップに合わせて、二つの OPEN 命令があります。
直接 FTDI 社のライブラリを呼び出す場合、シリアル通信チップですから、通信の速度 19200bps と　キャラクタ　8 ビット、stop=1bit 、ノンパリティ　を指定してください。
詳しくは FTDI 社のライブラリの解説書をご覧ください。

UM400DLL 解説(1) 専用 DLL ライブラリ命令

参考に、USB/RS232C インターフェイス上の FORMAT も記載しています。

USB インターフェイスの OPEN 処理(1) パラレル型 USB チップ製品の OPEN

```
int    LaboUSB_Open245(char* sname,  int unitNo,  int Rtimeout, int Wtimeout);
```

引数

sname	char	UM-400 の USB シリアル番号 8 文字を指定する
unitNo	int	通常は 1 を指定、複数台の UM-400 を USB に接続する場合は 1,2,3…と別々に番号を分けて OPEN します 後に続く命令は、ここで指定する番号を使って、UM-400 を区別しています。
Rtimeout	int	読出しの時のタイムアウトエラー時間を ms で指定
Wtimeout	int	書出しの時のタイムアウトエラー時間を ms で指定

戻り値: 正常終了ならば 0 エラーなら-1 を戻します

UM-400 はシリアル型 USB チップを使っていますから、この命令は使いません。

USB インターフェイスの OPEN 処理(2) シリアル型 USB チップ製品の OPEN

```
int    LaboUSB_Open232(char* sname,  int unitNo,
                       int ReadTimeOut,  int WriteTimeOut,  int Brate);
```

引数

sname	char	UM-400 の USB シリアル番号 8 文字を指定する
unitNo	int	通常は 1 を指定、複数台の UM-400 を USB に接続する場合は 1,2,3…と別々に番号を分けて OPEN します 後に続く命令は、ここで指定する番号を使って、UM-400 を区別しています。
Rtimeout	int	読出しの時のタイムアウトエラー時間を ms で指定
Wtimeout	int	書出しの時のタイムアウトエラー時間を ms で指定
Brate	Int	通信速度を指定します。 UM-400 の場合は 19200 を指定します。

戻り値: 正常終了ならば 0 エラーなら-1 を戻します

試験プログラムの実際の記述

```
rt=LaboUSB_Open232(INF.IDname ,1,500,500,19200);
```

USB インターフェイスを CLOSE する

```
int     LaboUSB_Close(int unitNo);
```

引数

unitNo	int	OPEN の時に指定した識別番号 1,2,3…を指定します。

戻り値: 正常終了ならば 0 エラーなら-1 を戻します

この命令は、プログラムの最後に呼び出します。

USB 通信の基本命令　WRITE

```
int　LaboUSB_Write(int unitNo,　char *data, int leng);
```

引数

unitNo	Int	OPEN の時に指定した識別番号 1,2,3…を指定します。
data	Char	出力する通信データの文字列
leng	int	出力する通信データのバイト数

戻り値: 正常終了ならば 0 エラーなら-1 を戻します

この命令は基本命令です。
装置の通信命令を直接、送り出す場合などに、利用しますが、通常は後述する
装置に特化した命令を使います。

USB 通信の基本命令　READ

```
int　LaboUSB_Read(int unitNo,　char *data,　int leng);
```

引数

unitNo	Int	OPEN の時に指定した識別番号 1,2,3…を指定します。
data	Char	読み取る通信データの文字列
leng	int	読み取る通信データのバイト数　（最大 4096）

戻り値: 正常終了ならば 0 エラーなら-1 を戻します

この命令は基本命令です。
装置の通信命令を直接、送り出す場合などに、利用しますが、通常は後述する
装置に特化した命令を使います。

USB 通信の基本命令　受信バッファデータ数の取得

```
int     LaboUSB_ReadQsize(int unitNo);
```

unitNo で指定する USB 装置からの受信バッファのデータ量を検査します。
バイト数がもどります。

UM-400 の通信データ数は、命令により固定になっています。
USB 装置からデータを受信する場合に、長さが不明な場合などに利用します。

USB 通信の基本命令　送受信バッファのサイズを指定する。

```
int     LaboUSB_RW_Qsize(int unitNo,  int *sizeR,  int *sizeW);
```

引数

unitNo	int	OPEN の時に指定した識別番号 1,2,3…を指定します。
sizeR	int	読み取る通信データの文字列
sizeW	int	読み取る通信データのバイト数

戻り値: 正常終了ならば 0 エラーなら-1 を戻します

USB ドライバ内のバッファサイズを変更します。
通常は使わなくても良いですが、通信データ量が大きい装置と接続する場合は
変更の必要があるかも知れません。
詳しくは FTDI 社のライブラリ仕様を確認してください。

UM-400 専用命令　　バージョン情報の読み出し

```
int    LaboUSB_UM_ReadVersion(int unitNo,  char *version);
```

引数

unitNo	int	OPEN の時に指定した識別番号 1,2,3…を指定します。
version	char	受信場所　　Version データは 8 バイトです

戻り値: 正常終了ならば 0 エラーなら-1 を戻します

実際のデータ FORMAT:

命令　３バイト固定

$	V	0x0D

応答　８バイト固定

R	E	V	1	.	2	3	0x0D

example

試験プログラム(C++Builder)では、読み出した Version 情報をステイタスバーへ表示しています。

```
StatusBar1->SimpleText="                              ";
LaboUSB_UM_ReadVersion(1,cbuf);
cbuf[8]=0x00;
StatusBar1->SimpleText=cbuf;
```

UM-400 専用命令　　デバッグモードの ON

```
int    LaboUSB_UM_DebugON(int unitNo);
```

unitNo で指定する UM-400 装置に送りだした命令を UM-400 本体の表示管に表示させることができます。

この命令は、UM-400 を使ったアプリケーションを開発している時 Windows 側のデバッカで、ステップ実行しながら UM-400 へ通信しているデータの確認をする場合に、便利です。

UM-400 がおかしいのではないか?
といった疑問が生じた場合は、実際に通信しているデータを表示させて見るのに便利です。

実際のデータ通信 FORMAT:
命令　４バイト固定

$	d	1	0x0D

この命令には、応答がありません。

UM-400 専用命令　　デバッグモードの OFF

```
int    LaboUSB_UM_DebugOFF(int unitNo);
```

unitNo で指定する UM-400 装置のデバッグモードを OFF にします。

実際のデータ通信 FORMAT:
命令　４バイト固定

$	d	0	0x0D

この命令には、応答がありません。

UM-400 専用命令　　モータの緊急停止

```
int    LaboUSB_UM_HaltMotor(int unitNo,  int Motor);
```

引数

unitNo	int	OPEN の時に指定した識別番号 1,2,3…を指定します。
Motor	int	モーター番号　1,2,3,4

戻り値: 正常終了ならば 0 エラーなら-1 を戻します

UM-400 のメインコントローラには、4 個のモータを接続できます。
モータ番号は、1,2,3,4 番です。

実際のデータ通信 FORMAT:
命令　4 バイト固定

$	H	☆	0x0D

☆は文字　‘1’‘2’‘3’‘4’ のどれか
この命令には、応答がありません。

すべてのモータをほ停止する場合は

```
rt=LaboUSB_UM_HaltMotor(1, 1);
rt=LaboUSB_UM_HaltMotor(1, 2);
rt=LaboUSB_UM_HaltMotor(1, 3);
rt=LaboUSB_UM_HaltMotor(1, 4);
```

のように 4 個のモータへ送ってください。

UM-400 専用命令　　モータ設定情報をハードウェアに記録

```
int    LaboUSB_UM_SaveParameter(int unitNo,  int Motor);
```

引数

unitNo	int	OPEN の時に指定した識別番号 1,2,3…を指定します。
Motor	int	モーター番号　1,2,3,4

戻り値: 正常終了ならば 0 エラーなら-1 を戻します

実際のデータ通信 FORMAT:

命令　４バイト固定

$	@	☆	0x0D

☆は文字　　‘1’ ‘2’ ‘3’ ‘4’ のどれか

この命令には、応答がありません。

リミッタスイッチを交換して、センサの極性をハード的に変更した場合

ソフトウェアで、リミッタの極性を指定したあと、この命令を送り出すと

UM-400 が極性を記憶します。

直接、UM-400 のボタン操作でも変更はできます。

通常は使わない命令です。

UM-400 専用命令　　データム処理

int　LaboUSB_UM_Datum(int unitNo,　int Motor,　char agu);

引数

unitNo	int	OPEN の時に指定した識別番号 1,2,3…を指定します。
Motor	int	モーター番号　1,2,3,4
agu	char	+　CW 方向のリミッタで停止 -　CCW 方向のリミッタで停止 P　CW or home リミッタで停止 M　CCW or home リミッタで停止

戻り値: 正常終了ならば 0 エラーなら-1 を戻します

実際のデータ通信 FORMAT:

命令　５バイト固定

$	D	☆	●	0x0D

☆は文字　‘1’ ‘2’ ‘3’ ‘4’ のどれか

●はデータムの種類 ‘+’ ‘-‘ ‘P’ ‘M’を指定する

この命令には、応答がありません。

機械原点サーチなどに利用します。

注意

データタム処理の直前に

LaboUSB_UM_HaltMotor(int unitNo,　int Motor);　を必ず実行してください。

UM-400 専用命令　　モーター回転速度の指定

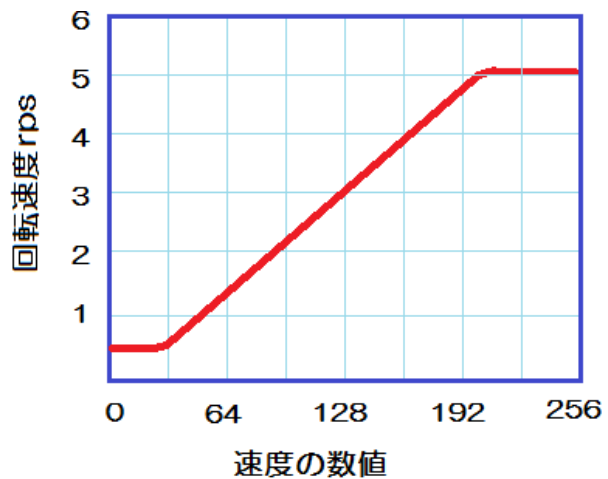
int　LaboUSB_UM_Speed(int unitNo,　int Motor, int speed);

引数

unitNo	int	OPEN の時に指定した識別番号 1,2,3…を指定します。
Motor	int	モーター番号　1,2,3,4
speed	int	速度を 1 から 150 で指定します。３２を加算した文字として指定

戻り値: 正常終了ならば０ エラーなら-1 を戻します

実際速度はグラフの様に変化します。

回転中でも指定可能です。

実際のデータ通信 FORMAT:

命令　５バイト固定

$	S	☆	●	0x0D

☆は文字　‘1’ ‘2’ ‘3’ ‘4’ のどれか

●はスペース文字に 1 から 150 を加算した文字で “!”#$%&’(),,,,012

この命令には、応答がありません。

UM-400 専用命令　　リミットの極性の設定

int　LaboUSB_UM_Limit(int unitNo,　int Motor,　char A,　char B);

引数

unitNo	int	OPEN の時に指定した識別番号 1,2,3…を指定します。
Motor	int	モーター番号　1,2,3,4
charA	char	リミッタの場所　ccw は　c’-‘　Home は c’H’　cw は c’+’
charB	char	リミッタの極性　c‘0’ は　負論理　c’1’ は正論理

戻り値: 正常終了ならば 0 エラーなら-1 を戻します

実際のデータ通信 FORMAT:

命令　6 バイト固定

$	P	☆	A	B	0x0D

☆は文字　‘1’ ‘2’ ‘3’ ‘4’ のどれか

A は　‘-‘　‘H’　‘+’のどれか

B は　‘0’　か　‘1’

この命令には、応答がありません。

リミッタスイッチの極性を指定します。

負論理とは、スイッチが、導通している時にモーターが回ります。

正論理とは、スイッチが、導通している時はモーターが回りません。

普通は、リミッタ線が断線したら、モータが停止する方が良いので負論理にしますが、リミッタを使わない場合は、正論理にすることもあります。

UM-400 専用命令　　絶対座標位置へ移動する

```
int    LaboUSB_UM_MoveABS(int unitNo,  int Motor, int address);
```

引数

unitNo	int	OPEN の時に指定した識別番号 1,2,3…を指定します。
Motor	int	モーター番号　1,2,3,4
address	int	移動したい座標を指定する　固定 9 文字の数値で指定する

戻り値: 正常終了ならば 0 エラーなら-1 を戻します

実際のデータ通信 FORMAT:

命令　14 バイト固定

$	A	☆	_	_	_	_	1	2	3	4	5	0x0D

☆は文字　　'1' '2' '3' '4' のどれか

9 文字の座標値

この命令には、応答がありません。

回転が速いと指定座標位置を超えて停止することがあります。
コントローラは停止座標位置を確認して座標正しい位置に再移動しますが
その間に座標移動命令が来ると、次の命令を実行します。
停止位置を確認して次の移動命令を実行するようにしましょう。

UM-400 専用命令　　指定した座標分、相対移動する

int　LaboUSB_UM_MoveREL(int unitNo,　int Motor,　int address);

引数

unitNo	int	OPEN の時に指定した識別番号 1,2,3…を指定します。
Motor	int	モーター番号　1,2,3,4
address	int	移動したい量を指定する　　固定 9 文字の数値で指定する

戻り値: 正常終了ならば 0 エラーなら-1 を戻します

実際のデータ通信 FORMAT:

命令　14 バイト固定 命令の文字はアルファベット O です。

$	O	☆	_	_	_	_	1	2	3	4	5	0x0D

☆は文字　‘1’‘2’‘3’‘4’ のどれか

9 文字の座標値

この命令には、応答がありません。

マイナス方向への移動は、マイナスの値を指定します

LaboUSB(1,1,-1000); は　モータ#1 を　CCW 方向へ 1000 パルス数戻します。

パルス数とは、エンコーダの増減パルス数です。

回転が速いと指定座標位置を超えて停止することがあります。

コントローラは停止座標位置を確認して座標正しい位置に再移動しますが

その間に座標移動命令が来ると、次の命令を実行します。

停止位置を確認して次の移動命令を実行するようにしましょう。

UM-400 専用命令　　座標値を変更します。

```
int    LaboUSB_UM_EditAddress(int unitNo,  int Motor, int address);
```

引数

unitNo	int	OPEN の時に指定した識別番号 1,2,3…を指定します。
Motor	int	モーター番号　1,2,3,4
address	int	新しい座標を指定します。

戻り値: 正常終了ならば 0 エラーなら-1 を戻します

LaboUSB_UM_EditAddress(1,1,0);　　　座標位置をリセットする
LaboUSB_UM_EditAddress(1,1,123);　　座標位置を 123 にする

実際のデータ通信 FORMAT:

命令　14 バイト固定

$	A	☆	_	_	_	_	1	2	3	4	5	0x0D

☆は文字　　‘1’ ‘2’ ‘3’ ‘4’ のどれか

9 文字の座標値

この命令には、応答がありません。

UM-400 専用命令　　座標値を読み込みます。

int　LaboUSB_UM_ReadAddress(int unitNo,　int Motor,　int* address);

引数

unitNo	int	OPEN の時に指定した識別番号 1,2,3…を指定します。
Motor	int	モーター番号　1,2,3,4
address	int	新しい座標を指定します。

戻り値: 正常終了ならば 0 エラーなら-1 を戻します

実際のデータ通信 FORMAT:

命令　４バイト固定

$	R	☆	0x0D

☆は文字　　'1' '2' '3' '4' のどれか

応答データ 10 文字

例　座標値が 123456 の時

_	_	_	1	2	3	4	5	6	0x0d

UM-400 専用命令　　モータの状態、リミッタ状態を読み出します。

```
int    LaboUSB_UM_ReadStatus(int unitNo,    int Motor,   int* status);
```

引数

unitNo	int	OPEN の時に指定した識別番号 1,2,3…を指定します。
Motor	int	モーター番号　1,2,3,4
status	int	モータのステイタスビットを読み込みます。

戻り値: 正常終了ならば 0 エラーなら-1 を戻します

リミッタの状態は 1 文字(8bit)です。

Bit-7	ERROR フラグ	一回でもエラーが発生すると ON
Bit-6	Time	
Bit-5	未使用	モータが回らないで止まった(負荷が大きい)
Bit-4	データム	データム中 ON
Bit-3	Busy	モータが回転中
Bit-2	ccw	ccw 方向のリミッタにぶつかると　ON
Bit-1	Home	Home リミッタにぶつかると　ON
Bit-0	cw	cw 方向のリミッタにぶつかると　ON

実際のデータ通信 FORMAT:

命令　４バイト固定

$	?	☆	0x0D

☆は文字　　‘1’ ‘2’ ‘3’ ‘4’ のどれか

応答 3 バイト固定

H	L	0x00

HL は８ビットの状態を２文字の HEX 文字

UM400DLL 解説(2) USB 命令

USB 命令とは、直接 UM-40M が受け付ける命令のことです。

USB 命令の FORMAT

UM-400M は、最大で 4 個のモータを接続できます。
USB からの命令の最初の 3 文字は

$	C	9	命令に対するデータで最後は CR で終わる

最初の文字は　c'$' です。
2 番目は命令コードです。
3 文字目には、モータ番号が入ります　文字の　c'1' c'2' c'3' c'4' のいずれかです。

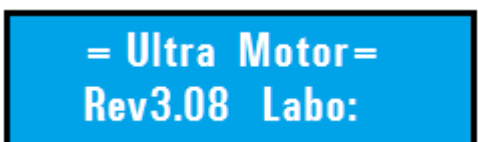
4 文字目以降は、命令に対する引数で、最後が CR コードで終わっている
必要かあります。

モータ 3 番の座標を-12345 へ変更する場合は
$W3　-12345<CR>と送ります。

USB / RS32C 回線上の命令内容

◆は文字の数字 1 文字で、1,2,3,4 のモータ番号

命令	文字列例	内容	応答
V	$V<CR>	バージョン情報読み込み	REV3.08<CR> 固定 8 文字
d	$D◆<CR>	デバッグモード ON/OFF	N=’1’なら ON ‘0’なら OFF
H	$H◆<CR>	指定モータの停止	
@	$@◆<CR>	指定モータの情報記録	
D	$D◆A<CR>	データム A=c’–‘ ccw 方向 A=c’+’ cw 方向 A=’Z’ home で停止 A=’P’ cw 方向+HOME A=’M’ ccw 方向れ HOME	
S	$S◆B<CR>	B=8bit の数値でモータの 回転速度を指定する 0x00 から 0xFF	
P	$P◆AB<CR>	リミッタの極性を指定 A=c’–‘　CCW リミッタ A=c’H‘　home リミッタ A=c’+‘　CW リミッタ B=’0’ 負論理接続 B=’1’ 正論理接続	
A	$A◆string<CR>	絶対座標への移動 String は文字列の数値で 座標位置を示す	String は 9 文字固定の数値です。
O	$R◆string<CR>	相対座標への移動 String は文字列の数値で 移動量を示す	String は 9 文字固定の数値です。
W	$W◆string<CR>	座標値の変更 String は文字列の数値で 座標値を示す	String は 9 文字固定の数値です。
?	$?◆<CR>	ステイタス読み込み	S<CR>　S=8bit 数値
R	$R◆<CR>	座標の読み出し	数字の文字列 9 ケタ<CR>が戻ります。

メンテナンス、初期化処理

ここでは、最初に超音波モーターを使う時の、ハードウェアの初期化方法や、モータの追加方法について、説明します。

電源を入れると

＝Ultra Motor＝
www.nabe-e.com

と表示され４秒程度で

＝Ultra Motor＝
Rev3.08 Labo:

と表示されます。

[SET]ボタンを押してください。

＝UM4000 SYSTEM＝
＝SETUP MODE OFF＝

と表示されます。

ＤＯＷＮまたはＵＰキーで表示を

＝UM4000 SYSTEM＝
＝SETUP MODE ON!＝

にしてください。

これで、メンテナンスができるようになりました。

UM-400P ボードを初期化する

新しくモータを追加する場合は、ＵＭ－４００Ｐ基板を追加します。
納入時、ＵＭ－４００Ｐ基板は初期化されています。
初期化の内容は、いろいろなパラメータの初期値を書き込むことです。

初期値の設定方法
［ＭＥＮＵ］ボタンを数回押しながら

Motor　Select ....
MOTOR=　3 (1..4)

Ｍｏｔｏｒ　Ｓｅｌｅｃｔ画面にして、Ｄｏｗｎ／ＵＰキーで
モーター番号１，２，３，４を設定します。
次に［ＭＥＮＵ］ボタンを数回押しながら、

[ﾒﾝﾃﾅﾝｽ nabe-e]
PowerON　110

メンテナンスメニューを選択して、Ｄｏｗｎ／ＵＰキーで、
M#=2 INIT EEPROM
を表示させて　［ＳＥＴ］ボタンを押します。
上の例ではモータ２番のＵＭ－４００Ｐ基板をょキ科しています。

ＵＭ－４００リミッタの極性を指定する

Motor　Select ....

MOTOR=　3 (1..4)

メニューでＤｏｗｎ／ＵＰキーを使い、モータ軸１，２，３，４のどれかを選択します。
メニューでＬｉｍｉｔ極性の指定画面にします。

Limit　 Polarity..

M#=　3 CW=NEG

Ｄｏｗｎ／ＵＰキーを使い
CW=NEG　CW=POS のいずれかの表示で［ＳＥＴ］を押す
CCW=NEG　CW=POS のいずれかの表示で［ＳＥＴ］を押す
Hom=NEG　Hom=POS のいずれかの表示で［ＳＥＴ］を押す
のようにして、リミッタの極性を指定します。

最後に［ＭＥＮＵ］キーを押して

SAVE　Parameter

[SET] is execute

を表示させて、［ＳＥＴ］ボタンを押すと、終わりです。

制御モータ数の変更

ＵＭ－４００Ｐは、個々のモータの初期化でした。
モータ数の変更は　ＵＭ－４００Ｐ基板ではなく、ＵＭ－４００M基板に
何枚のモータコントローラボードを接続しているかを教えます。
接続されていないハードウェアをアクセスするとタイムアウト処理などで効率が
悪くなるからです。

＝UM4000 SYSTEM＝
＝SETUP　MODE ON!＝

にしてください。

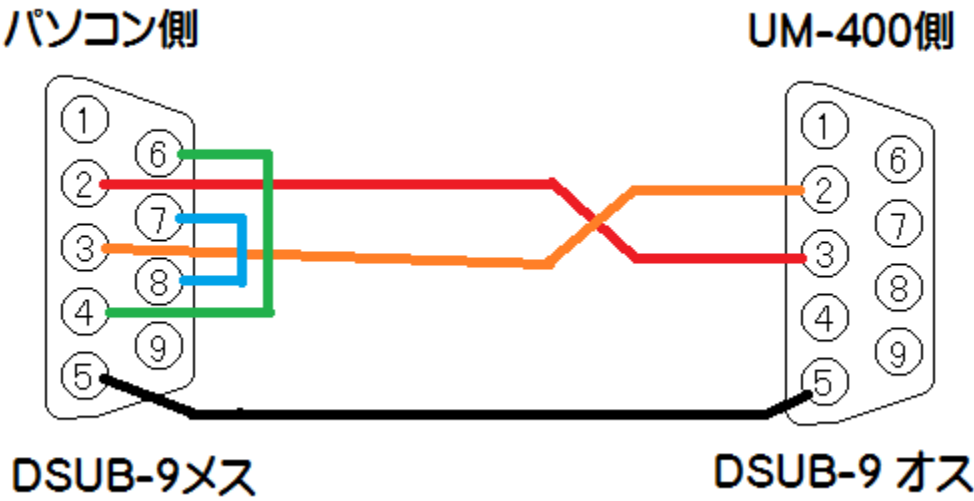
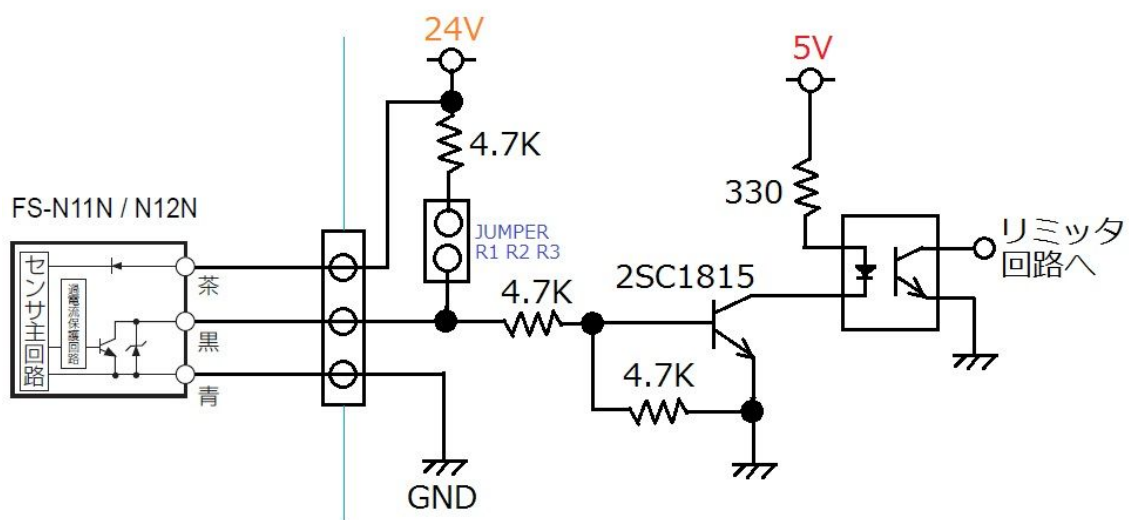
［ＭＥＮＵ］キーを押して、メンテナンスメニューを表示させて
[ﾒﾝﾃﾅﾝｽ　nabe-e]
Max motor　1 -> 3
のように　Ｍａｘ　ｍｏｔｏｒ指定を選びます。
Ｄｏｗｎ／ＵＰキーを使い
現在の軸数が１だとして

1->2
1->3
1->4

の軸数２，３，４のどれかを選択して［ＳＥＴ］ボタンを押します。
１軸から３軸へ変更する場合　１－＞３の表示で［ＳＥＴ］ボタンを押します
表示が ３－＞３になります。

その他

ケース製品のケースサイズ	W=323 H=138 D=280　取っ手を含まず
ケース製品の消費電流	100V 1A 以下　(13W/USR30　30W/USR60)

RS232C 接続ケーブルの作成
　UM-400 は USB 接続ですが、注文時に RS232C を選択できます。
19200 bps 8bit stop-1 パリティ無　で接続します。

オープンコレクタ出力タイプのセンサの接続例

モータコントローラのジャンパー R1,R2,R3 を短絡して使います。